TDD-övning: Deklaration

Så här i deklarationstider ska vi överta Skatteverkets uppgift och skapa ett program för e-deklaration. Den faktiska beräkningen är komplicerad, så för att förenkla uppgiften lite grann begränsas den till uträkning av skatt baserat på inkomst och kommun.

Målet med övningen är att testdriva fram ett program som tar två parametrar: inkomst (en siffra) och kommun (en sträng), och som beräknar skatten baserat på affärsreglerna nedan.

Beroende på din erfarenhet av testdriven utveckling kan du välja bland olika variationer av övningen:

Testdriv en klass/modul

Om du testar på testdriven utveckling för första gången kan det vara bra ett försöka sig på att testdriva fram en klass eller modul med motsvarande gränssnitt:

```
double finalTax = TaxCalculator.ComputeTax(300000, "Stockholm");
```

Testdriv en interaktiv applikation

Gör applikationen interaktiv på något sätt: konsol-, webb-applikation eller fönsterapplikation. En konsolapplikation kan fungera som följande:

```
EdeklarationConsole.exe 300000 Stockholm
```

Testdriv en interaktiv application med London-skolan

Gör applikationen interaktiv på något sätt: konsol-, webb-applikation eller fönsterapplikation, och använd London-skolans tillvägagångssätt. Börja närmast användargränsnittet och använd stubs och mock-objekt medan du jobbar dig inåt mot programelementen där affärslogiken finns. Om du använder C# och skriver en konsolapplikation kan du med smärre modifieringar av klassen `TaxCalculatorEndToEndTest` få till ett end-to-end-test.

Strikt TDD med hjälp av TPP

Välj gränssnitt och hur programmet ska startas och testdriv applikationen baserat på ”Transformation Priority Premise” (https://8thlight.com/blog/uncle-bob/2013/05/27/TheTransformationPriorityPremise.html).
Eftersom du kommer att spendera en hel del tid med att läsa på i denna övning är det troligt att du endast kommer att hinna med ett fåtal tester. Var beredd att dela med dig av dina observationer till resten av gruppen dock.

Affärsregler

1. Det finns ett grundavdrag på 13 400 kr. Inkomster under denna gräns beskattas inte.
2. Man betalar olika kommunalskatt beroende på vilken kommun man bor i. En del kommunala skattesatser finns i `Skattesatser.pdf`
3. Det finns två skiktgränser. Den nedre skiktgränsen är 455 300 kronor. På inkomster över denna gräns tas statlig inkomstskatt ut med 20 procent. På grund av grundavdraget betalas skatten effektivt på inkomster över 468 700 (455 300 + 13 400) kr.
4. Den övre skiktgränsen är 662 300 kronor. På inkomster över denna gräns tas statlig inkomstskatt ut med ytterligare 5 procent (den så kallade värnskatten), det vill säga sammanlagt 25 procent. Åter igen blir den faktiska gränsen 675 700 (662 300 + 13 400) kr.

För att övningen inte ska bli en räkne- och tolkningsövning finns ett antal exempel bifogade. Se `Exempel.xlsx` eller `Exempel.pdf`.

Tips och förslag

- ✓ Glöm inte att slica upp funktionaliteten och leverera iterativt och i inkrement. Det är mer belönande att producera en lösning som löser ett fall korrekt än att ha nånting som teoretiskt löser hela problemet men inte kompilerar.
- ✓ Fastna inte i input/output. Resultatet av beräkningen måste presenteras, men felhanteringen ska sparas till sist.
- ✓ Olika locales kan ställa till det med olika decimalavskiljare. Fastna inte i detta.
- ✓ Glöm inte Single Responsibility Principle, speciellt om du kör London-skolan.

Tips till dig som använder mockist-tdd (London-skolan)

Programmet är tämligen algoritmiskt till sin natur, så programelementen där beräkningarna sker kommer att behöva gamla hederliga asserts. Dock finns det utrymme för objekt som samarbetar med varandra:

- ✓ Någonting behöver ta indata från användaren
- ✓ Någonting gör beräkningen
- ✓ Någonting känner till de olika kommunerna och deras skattesatser

Utmaningen är således att klura ut hur många klasser/moduler som behövs och hur de samarbetar med varandra.